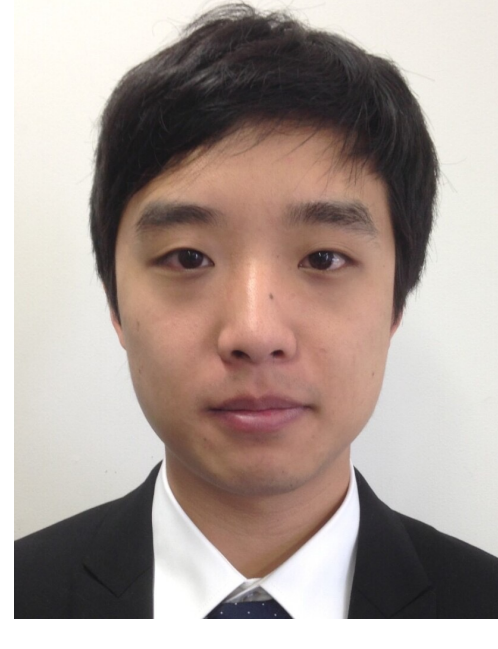


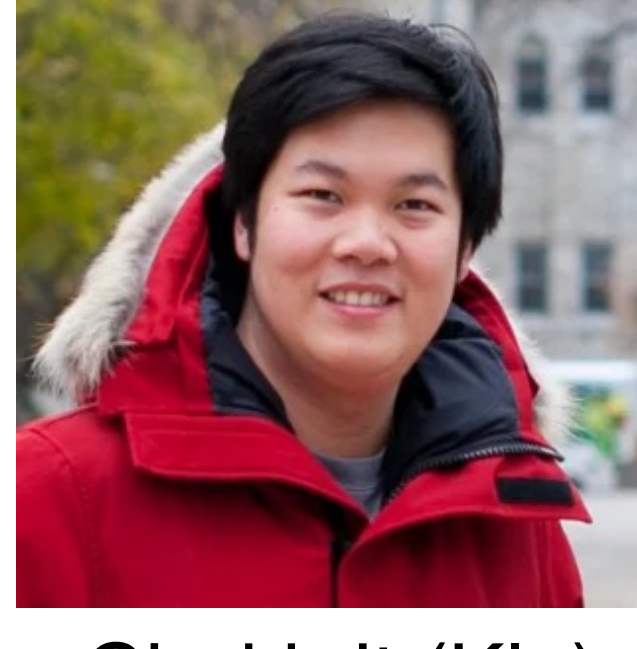
Where Should I Look at? Recommending Lines that Reviewers Should Pay Attention To



MONASH University



Yang Hong
Monash University, Australia



Chakkrit (Kla)
Tantithamthavorn
Monash University, Australia



Patanamon (Pick)
Thongtanunam
The University of Melbourne, Australia



THE UNIVERSITY OF
MELBOURNE

Summary

- Modern code review is an effective quality assurance practice, yet could be time-consuming to carefully review all new code in a patch.
- We propose REVSPOT—a machine learning-based approach to help reviewers to reduce their reviewing effort by only reviewing a smaller set of lines, increasing code review speed and reviewers' productivity.
- REVSPOT can accurately predict problematic lines (i.e., lines that will receive comments and will be revised) with a Top-10 Accuracy of 81% and 93%, which is 56% and 15% better than the baseline approach using N-gram.
- The majority of problematic lines that REVSPOT can correctly predict are related to *logic defects*.

REVSPOT—AN APPROACH TO RECOMMEND LINES THAT REVIEWERS SHOULD PAY ATTENTION TO

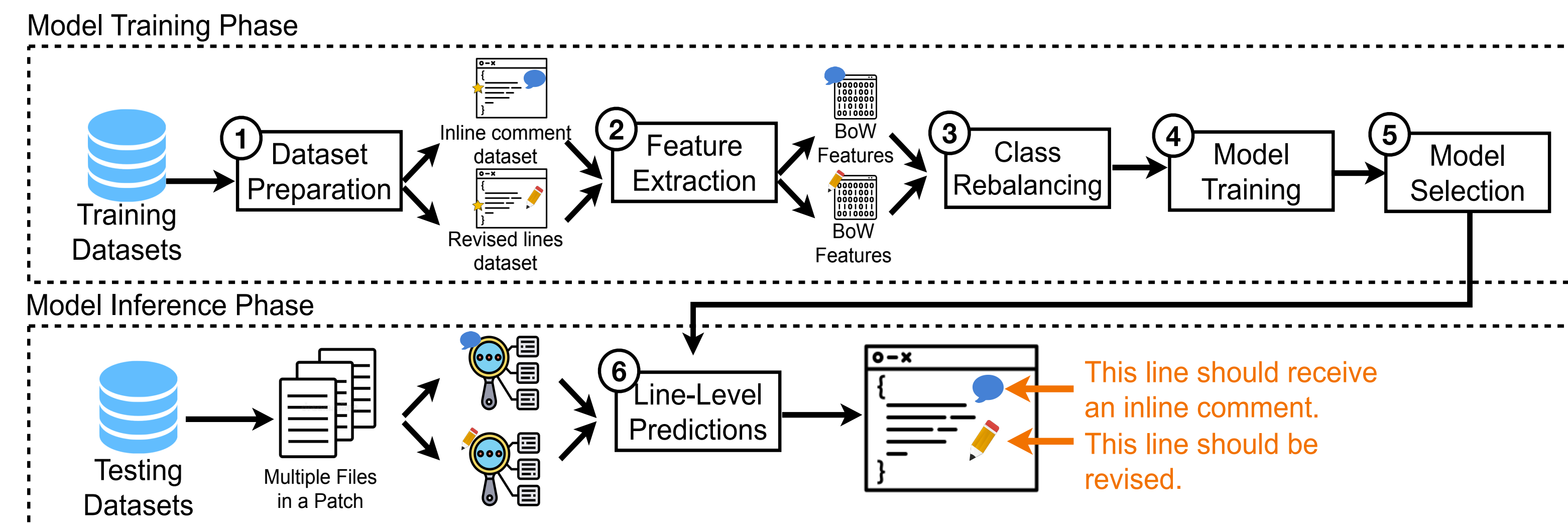


Figure 3. An overview of REVSPOT approach

Background

Steps for Modern Code Review (Figure 1):

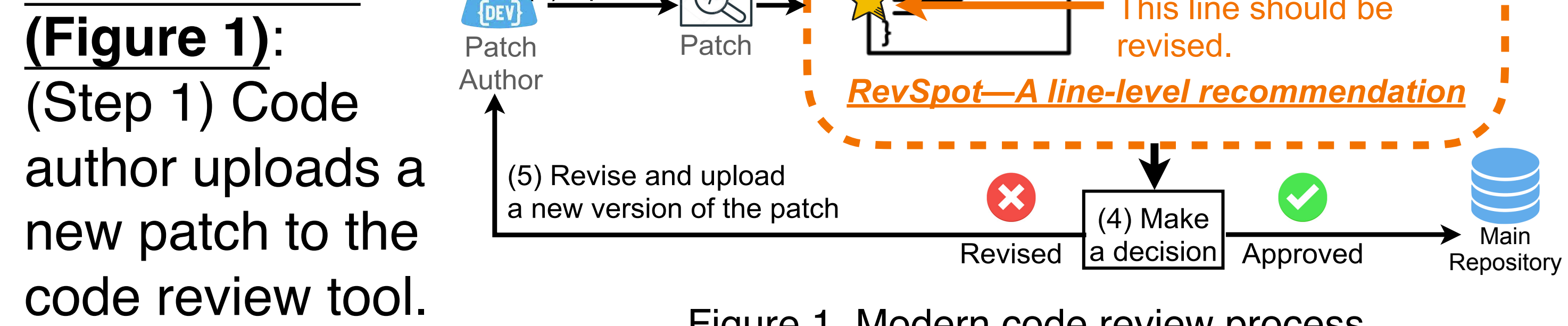


Figure 1. Modern code review process

- (Step 2) Patch author invites reviewers to examine the changed code in the proposed patch.
- (Step 3) Reviewers review the changed code. If the reviewers find problems or have concerns, they can provide comments to specific lines of code.
- (Step 4) Reviewers decide whether this patch can be integrated into the main code repository.
- (Step 5) If reviewers reject the patch, code author may revise the patch. If reviewers approve the patch, the patch will be integrated into the main repository.

Problem Motivation

- Since code review involves manual works from developers, qualitative studies reported that managing time is the top challenge faced by developers when performing code reviews.
- Patch authors may also spend unnecessary time to wait for reviewer feedback.

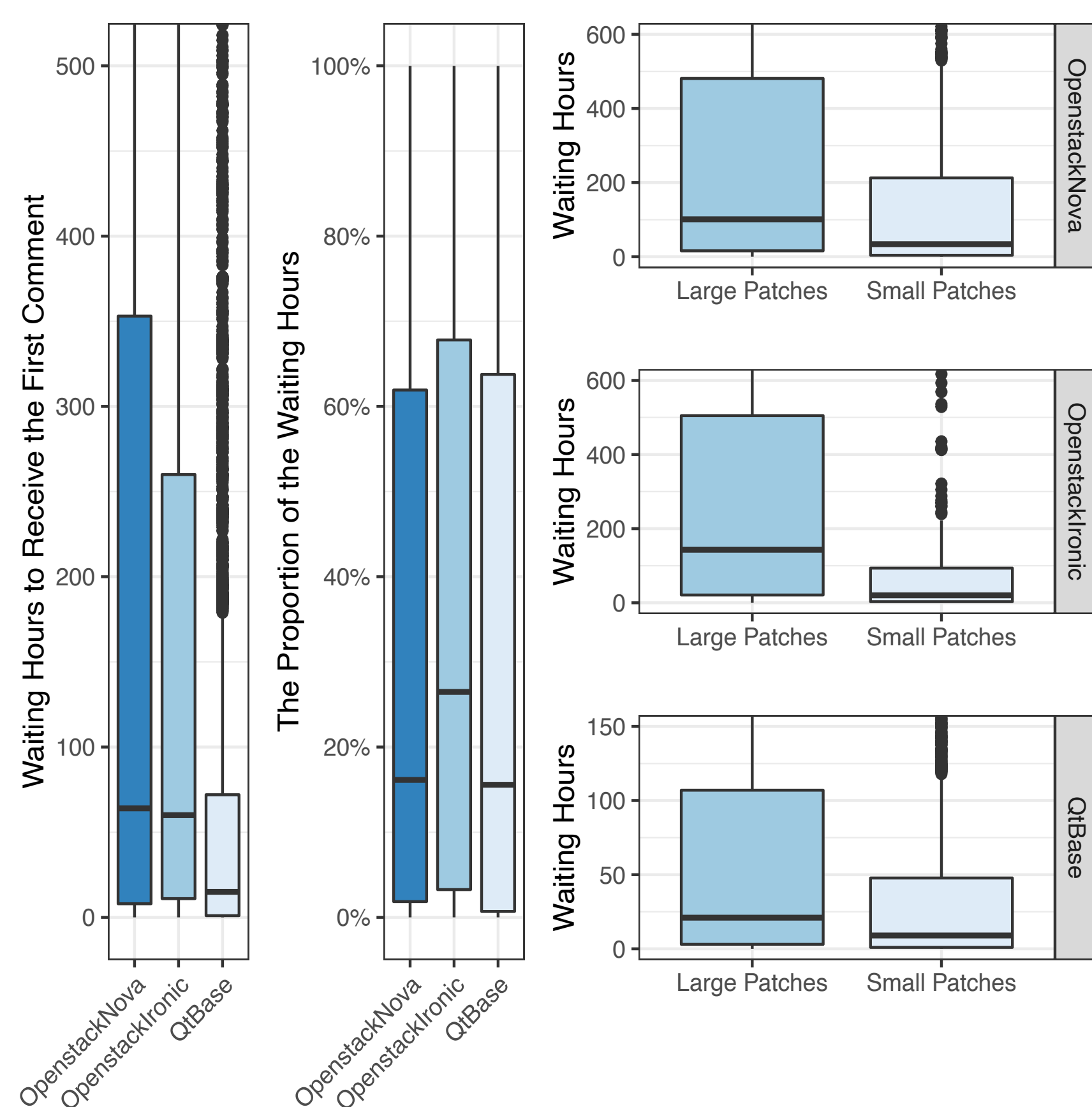


Figure 2. The waiting time to receive the first feedback (left), its proportion for the total reviewing time (middle), and the waiting time between large patches and small patches (right).

RQ1: How long do the patch authors wait to receive initial feedback from reviewers?

- Patch authors wait 15-64 hours to receive initial feedback from reviewers, which accounts for 16%-26% of the whole code review time of a patch.
- Larger patches tend to receive the initial feedback from reviewers slower than smaller patches.

Approach (Figure 3):

REVSPOT first converts the file content into the **Bag-of-Words (BOW)** features. Then, to rebalance the dataset, REVSPOT employs a Synthetic Minority Oversampling technique (**SMOTE**). After that, REVSPOT builds a file-Level model using **Random Forest**. Finally, to predict the lines that reviewers should pay attention to, REVSPOT compute the important score of each token feature for each file using a Local Interpretable Model-agnostic Explanations (**LIME**) and select top-10 tokens based on the descending order of the token important scores. Lines are ranked based on the number of top-10 important tokens, i.e., the more top-10 important tokens that lines have, the higher rank of the lines.

RQ2: How accurate is REVSPOT in predicting problematic lines that reviewers should pay attention to?

For predicting lines that will receive comments, REVSPOT is 56% (Top-10 accuracy) and 15% (d2h) better than the N-gram, indicating that REVSPOT outperforms N-gram approach.

For predicting lines that will be revised, REVSPOT is 15% (Top-10 accuracy) and 15% (d2h) better than the N-gram, indicating that REVSPOT outperforms N-gram approach.

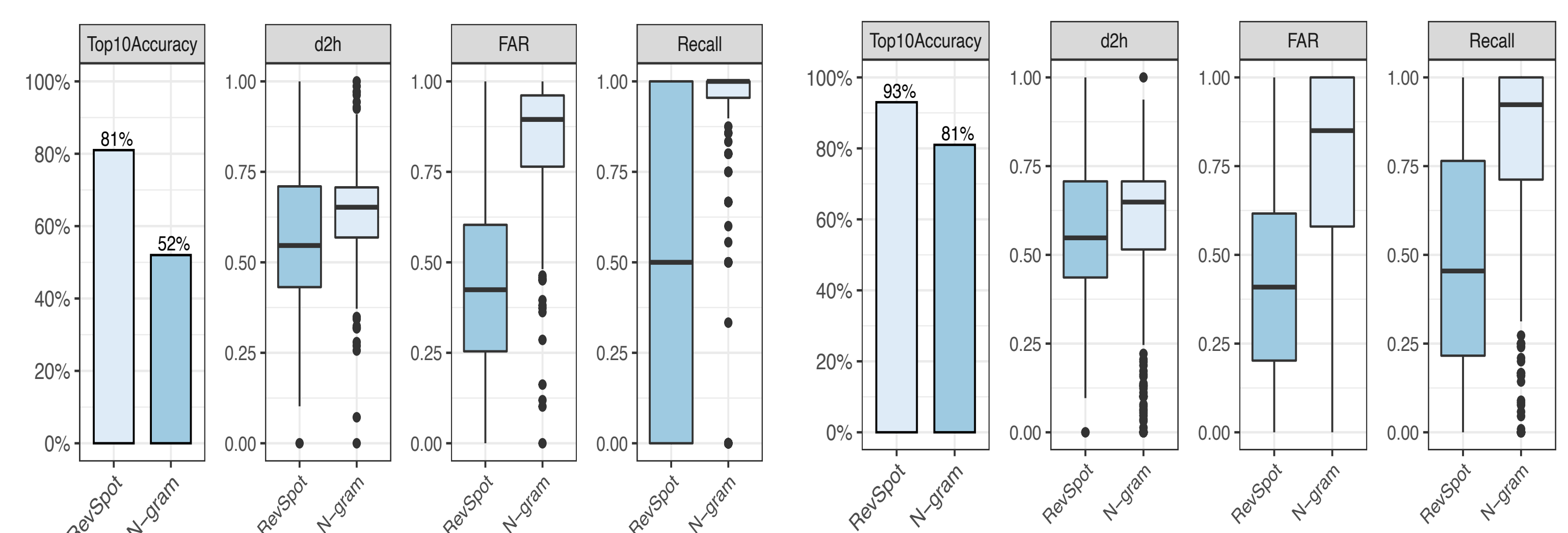


Figure 4. The accuracy of line-level prediction of REVSPOT to predict lines that will receive comments (left) and lines that will be revised (right).

RQ3: What kinds of defects in the problematic lines that are correctly predicted and incorrectly predicted?

- We manually categorize the samples of problematic lines into five defects (i.e., Logic defect, Interface defects, Structure defects, Documentation defects and Visual Defects).
- The majority of problematic lines that REVSPOT can correctly predict are related to logic defects.
- REVSPOT can correctly predict problematic lines that could impact the functionality of the system.

Acknowledgement

Chakkrit Tantithamthavorn and Patanamon Thongtanunam were supported by the Australian Research Council's Discovery Early Career Researcher Award (DECRA) funding scheme (DE200100941 and DE210101091).